Fig. 1

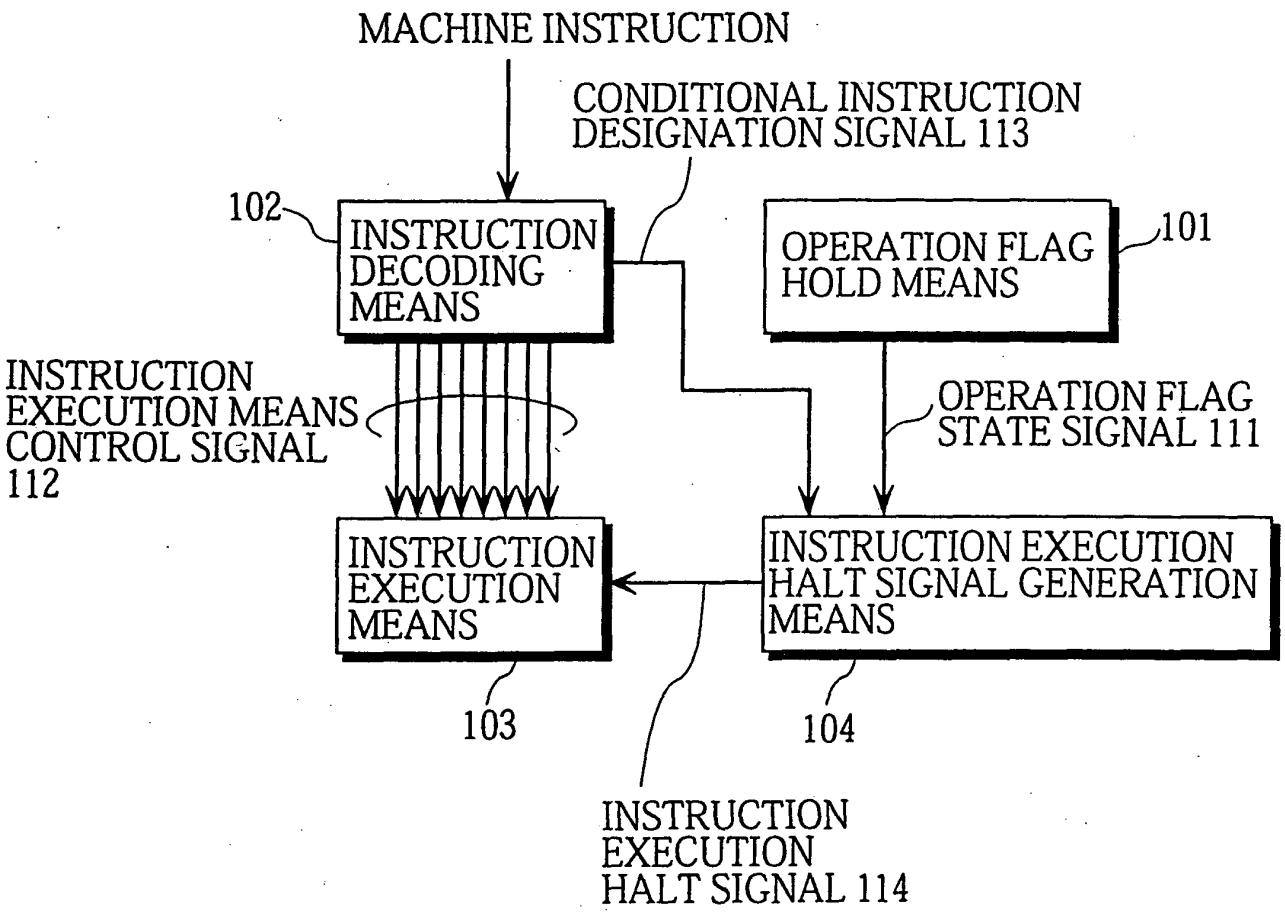
MACHINE INSTRUCTION
CONDITIONAL INSTRUCTION DESIGNATION SIGNAL 113
102
INSTRUCTION DECODING MEANS
OPERATION FLAG HOLD MEANS
101
INSTRUCTION EXECUTION MEANS CONTROL SIGNAL 112
OPERATION FLAG STATE SIGNAL 111
INSTRUCTION EXECUTION MEANS
INSTRUCTION EXECUTION HALT SIGNAL GENERATION MEANS
103
104
INSTRUCTION EXECUTION HALT SIGNAL 114

Fig. 2

CONDITIONAL TRANSFER INSTRUCTION 201			CONDITION 202
moveq	←	203	=
movgt	←	204	>
movge	←	205	≧

Fig. 3

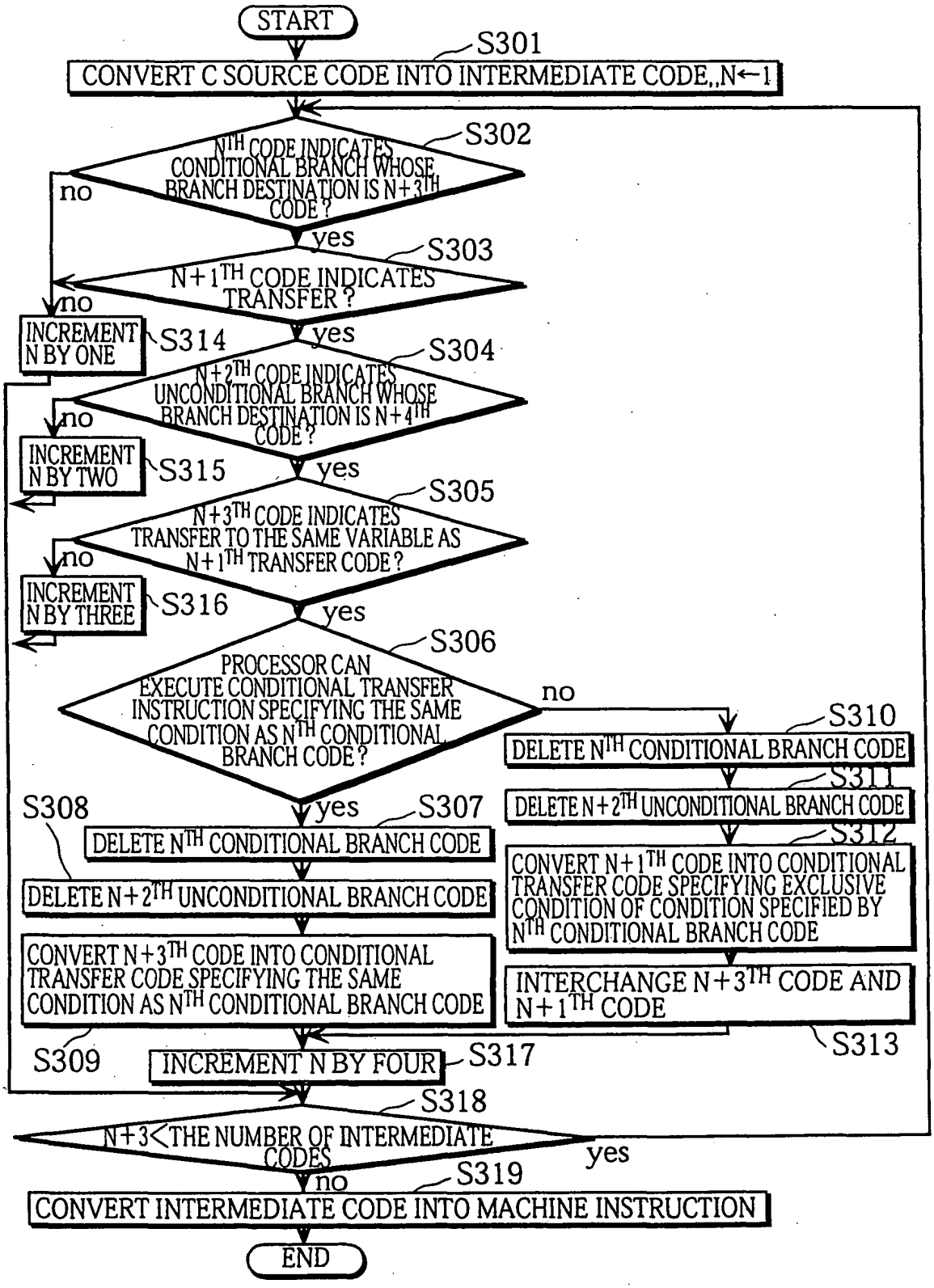
START
S301
CONVERT C SOURCE CODE INTO INTERMEDIATE CODE,,N←1
S302
N^TH CODE INDICATES CONDITIONAL BRANCH WHOSE BRANCH DESTINATION IS N+3^TH CODE?
no
yes
S303
N+1^TH CODE INDICATES TRANSFER?
no
yes
INCREMENT N BY ONE
S314
S304
N+2^TH CODE INDICATES UNCONDITIONAL BRANCH WHOSE BRANCH DESTINATION IS N+4^TH CODE?
no
yes
INCREMENT N BY TWO
S315
S305
N+3^TH CODE INDICATES TRANSFER TO THE SAME VARIABLE AS N+1^TH TRANSFER CODE?
no
yes
INCREMENT N BY THREE
S316
S306
PROCESSOR CAN EXECUTE CONDITIONAL TRANSFER INSTRUCTION SPECIFYING THE SAME CONDITION AS N^TH CONDITIONAL BRANCH CODE?
no
yes
S310
DELETE N^TH CONDITIONAL BRANCH CODE
S311
DELETE N+2^TH UNCONDITIONAL BRANCH CODE
S312
CONVERT N+1^TH CODE INTO CONDITIONAL TRANSFER CODE SPECIFYING EXCLUSIVE CONDITION OF CONDITION SPECIFIED BY N^TH CONDITIONAL BRANCH CODE
INTERCHANGE N+3^TH CODE AND N+1^TH CODE
S313
S308
S307
DELETE N^TH CONDITIONAL BRANCH CODE
DELETE N+2^TH UNCONDITIONAL BRANCH CODE
CONVERT N+3^TH CODE INTO CONDITIONAL TRANSFER CODE SPECIFYING THE SAME CONDITION AS N^TH CONDITIONAL BRANCH CODE
S309
INCREMENT N BY FOUR
S317
S318
N+3<THE NUMBER OF INTERMEDIATE CODES
yes
no
S319
CONVERT INTERMEDIATE CODE INTO MACHINE INSTRUCTION
END

Fig. 4A

```
if(a == b)
{
        c = 1;
}
else
{
        c = 0;
}
f();
```

Fig. 4B

```
if(a != b)
{
        c = 1;
}
else
{
        c = 0;
}
f();
```

Fig. 5A

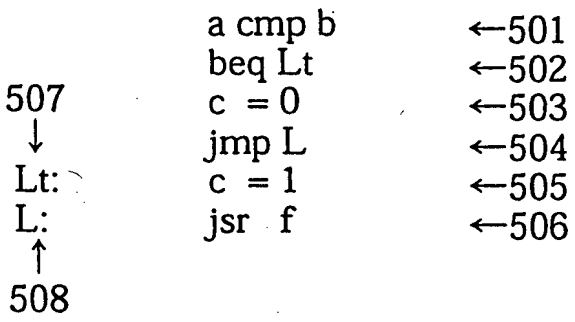
a cmp b
beq Lt
c = 0
jmp L
c = 1
jsr f
←501
←502
←503
←504
←505
←506
507
Lt:
L:
508

Fig. 5B

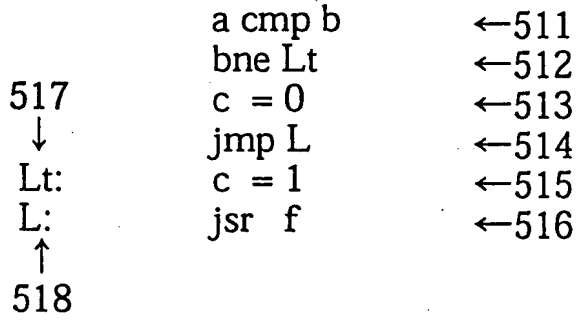
a cmp b
bne Lt
c = 0
jmp L
c = 1
jsr f
←511
←512
←513
←514
←515
←516
517
Lt:
L:
518

Fig. 6A

```
a cmp b          ←601
c =  0           ←602
c = :eq 1        ←603
jsr  f           ←604
```

Fig. 6B

```
a cmp b          ←611
c =  1           ←612
c = :eq 0        ←613
jsr  f           ←614
```

Fig. 7A

```
cmp      r0,r1     ←701
mov      0,r2      ←702
moveq    1,r2      ←703
jsr      f         ←704
```

Fig. 7B

```
cmp      r0,r1     ←711
mov      1,r2      ←712
moveq    0,r2      ←713
jsr      f         ←714
```

Fig. 8

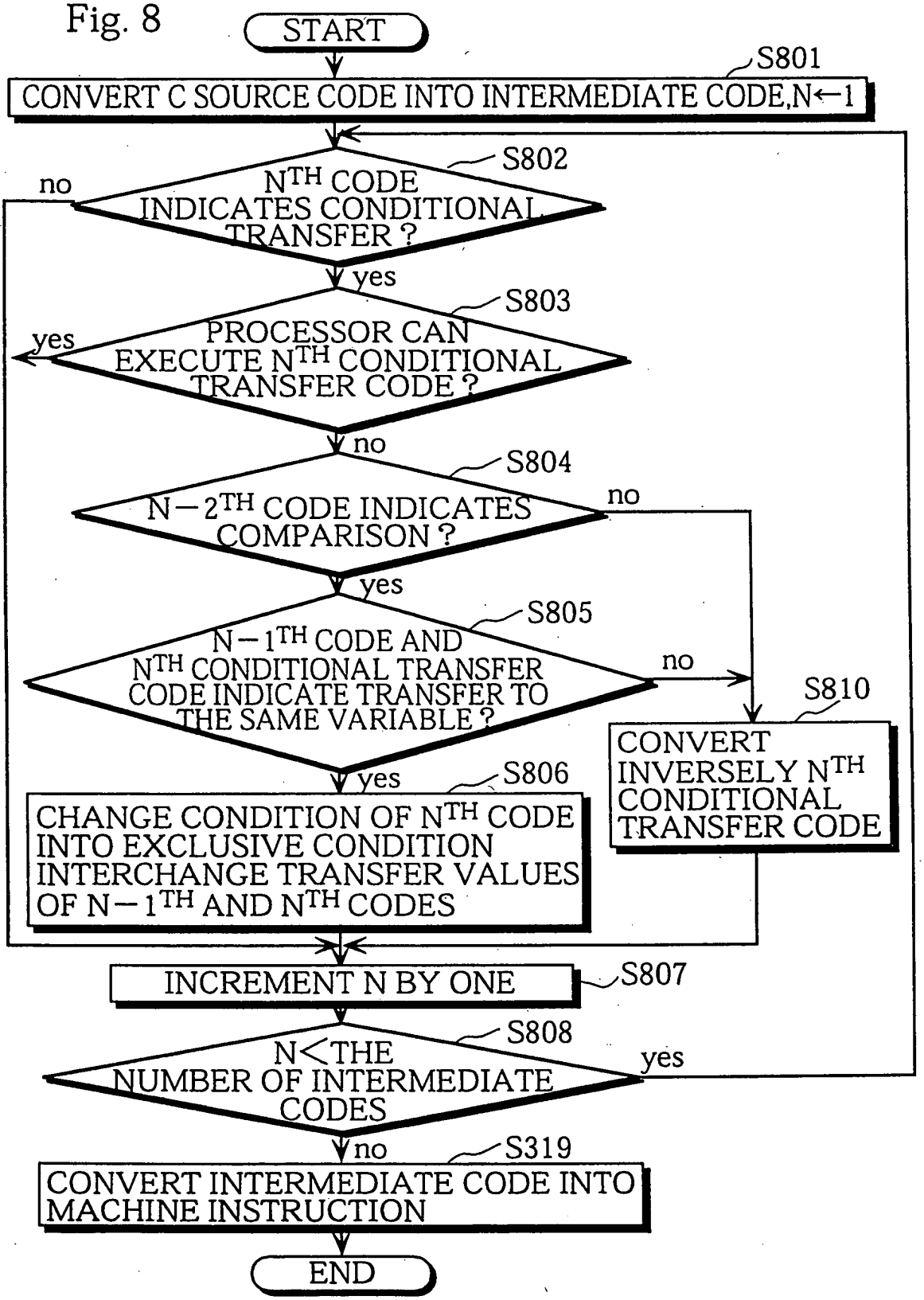
START
S801
CONVERT C SOURCE CODE INTO INTERMEDIATE CODE,N←1
S802
Nth CODE INDICATES CONDITIONAL TRANSFER ?
no
yes
S803
PROCESSOR CAN EXECUTE Nth CONDITIONAL TRANSFER CODE ?
yes
no
S804
N−2TH CODE INDICATES COMPARISON ?
no
yes
S805
N−1TH CODE AND Nth CONDITIONAL TRANSFER CODE INDICATE TRANSFER TO THE SAME VARIABLE ?
no
yes
S806
CHANGE CONDITION OF Nth CODE INTO EXCLUSIVE CONDITION INTERCHANGE TRANSFER VALUES OF N−1TH AND Nth CODES
S810
CONVERT INVERSELY Nth CONDITIONAL TRANSFER CODE
INCREMENT N BY ONE
S807
S808
N<THE NUMBER OF INTERMEDIATE CODES
yes
no
S319
CONVERT INTERMEDIATE CODE INTO MACHINE INSTRUCTION
END

Fig. 9

```
a cmp b          ←901
c =  0           ←902
c = :ne 1        ←903
jsr  f           ←904
```

Fig. 10

CONDITIONAL BRANCH INSTRUCTION 1001		CONDITION 1002
beq	←1003	=
bgt	←1004	>
bge	←1005	≧

Fig. 11

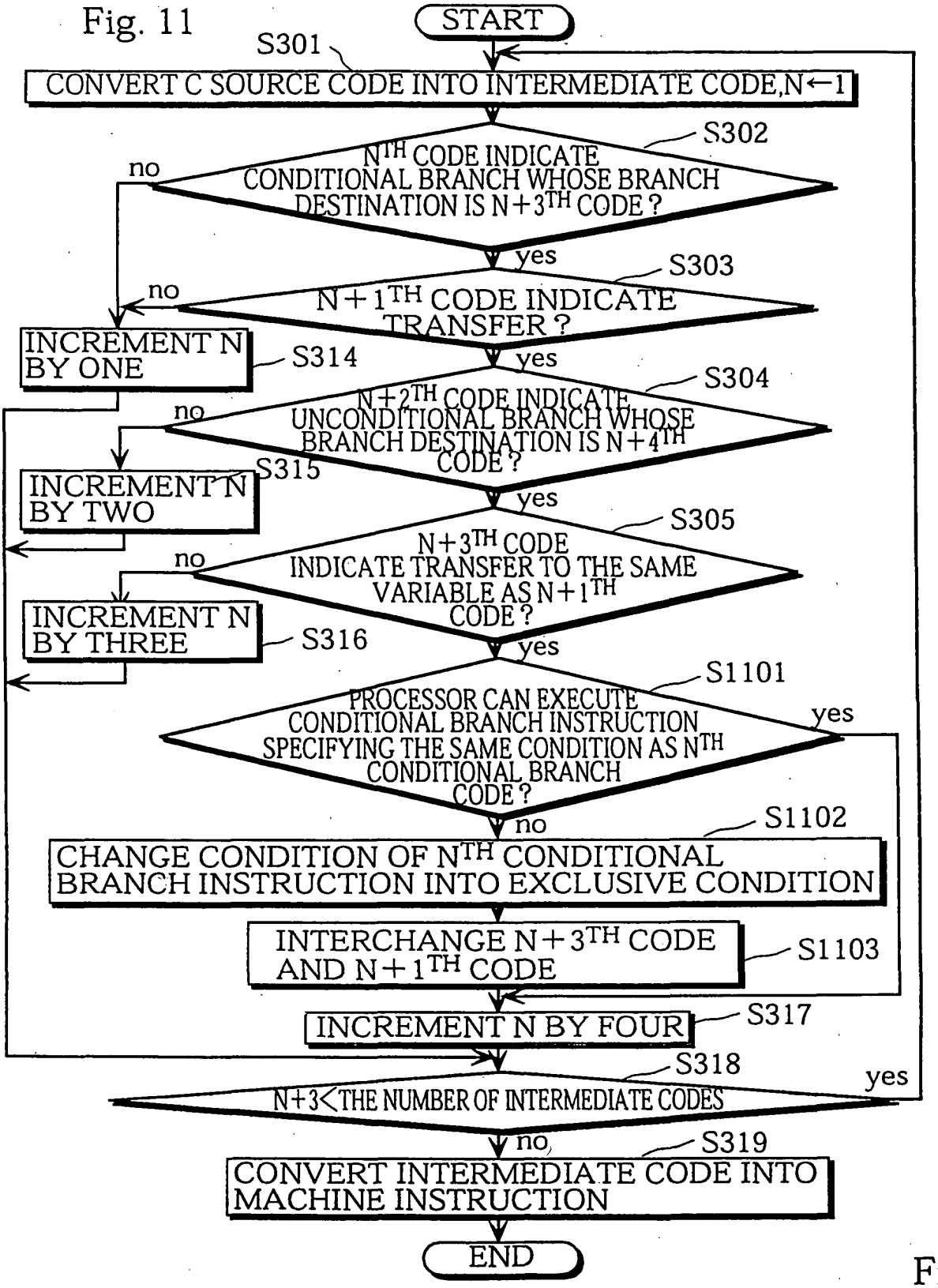
START
S301
CONVERT C SOURCE CODE INTO INTERMEDIATE CODE,N←1
S302
N^TH CODE INDICATE CONDITIONAL BRANCH WHOSE BRANCH DESTINATION IS N+3^TH CODE ?
no
yes
S303
N+1^TH CODE INDICATE TRANSFER ?
no
yes
INCREMENT N BY ONE
S314
S304
N+2^TH CODE INDICATE UNCONDITIONAL BRANCH WHOSE BRANCH DESTINATION IS N+4^TH CODE ?
no
yes
INCREMENT N BY TWO
S315
S305
N+3^TH CODE INDICATE TRANSFER TO THE SAME VARIABLE AS N+1^TH CODE ?
no
yes
INCREMENT N BY THREE
S316
S1101
PROCESSOR CAN EXECUTE CONDITIONAL BRANCH INSTRUCTION SPECIFYING THE SAME CONDITION AS N^TH CONDITIONAL BRANCH CODE ?
yes
no
S1102
CHANGE CONDITION OF N^TH CONDITIONAL BRANCH INSTRUCTION INTO EXCLUSIVE CONDITION
INTERCHANGE N+3^TH CODE AND N+1^TH CODE
S1103
INCREMENT N BY FOUR
S317
S318
N+3<THE NUMBER OF INTERMEDIATE CODES
yes
no
S319
CONVERT INTERMEDIATE CODE INTO MACHINE INSTRUCTION
END

Fig. 1

Fig. 12

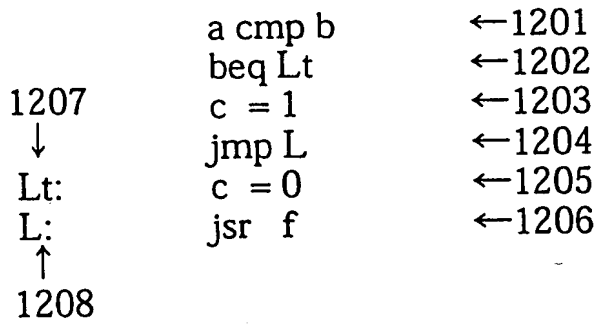
a cmp b
beq Lt
1207
c = 1
jmp L
Lt:
c = 0
L:
jsr f
1208
←1201
←1202
←1203
←1204
←1205
←1206

Fig. 13

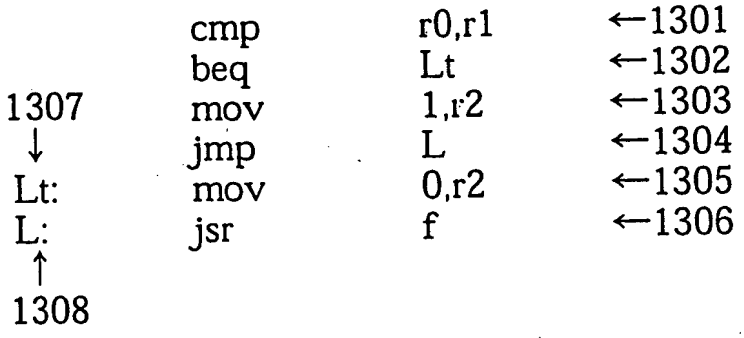
cmp
r0,r1
beq
Lt
1307
mov
1,r2
jmp
L
Lt:
mov
0,r2
L:
jsr
f
1308
←1301
←1302
←1303
←1304
←1305
←1306

Fig. 14

INSTRUCTION IN MNEMONIC CODE [COMPARISON]	SPECIFIED CONDITION	OPERATION
cmpeq Rm,Rn	=	SET CODITIONAL FLAG WHEN RESULT OF CMP INSTRUCTION INDICATE Rm AND Rn ARE EQUAL,OTHERWISE RESET CONDITIONAL FLAG
cmpne Rm,Rn	≠	SET CODITIONAL FLAG WHEN RESULT OF CMP INSTRUCTION INDICATE Rm AND Rn ARE NOT EQUAL,OTHERWISE RESET CONDITIONAL FLAG
cmpge Rm,Rn	≧	SET CODITIONAL FLAG WHEN RESULT OF CMP INSTRUCTION INDICATE Rm IS GREATER THAN OR EQUAL TO Rn AS DATA WITH SIGNS,OTHERWISE RESET CONDITIONAL FLAG
cmple Rm,Rn	≦	SET CODITIONAL FLAG WHEN RESULT OF CMP INSTRUCTION INDICATE Rm IS SMALLER THAN OR EQUAL TO Rn AS DATA WITH SIGNS,OTHERWISE RESET CONDITIONAL FLAG
cmpgt Rm,Rn	>	SET CODITIONAL FLAG WHEN RESULT OF CMP INSTRUCTION INDICATE Rm IS GREATER THAN Rn AS DATA WITH SIGNS,OTHERWISE RESET CONDITIONAL FLAG
cmplt Rm,Rn	<	SET CODITIONAL FLAG WHEN RESULT OF CMP INSTRUCTION INDICATE Rm IS SMALLER THAN Rn AS DATA WITH SIGNS,OTHERWISE RESET CONDITIONAL FLAG
cmpns Rm,Rn	≧	SET CODITIONAL FLAG WHEN RESULT OF CMP INSTRUCTION INDICATE Rm IS GREATER THAN OR EQUAL TO Rn AS DATA WITHOUT SIGNS,OTHERWISE RESET CONDITIONAL FLAG
cmpls Rm,Rn	≦	SET CODITIONAL FLAG WHEN RESULT OF CMP INSTRUCTION INDICATE Rm IS SMALLER THAN OR EQUAL TO Rn AS DATA WITHOUT SIGNS,OTHERWISE RESET CONDITIONAL FLAG
cmphi Rm,Rn	>	SET CODITIONAL FLAG WHEN RESULT OF CMP INSTRUCTION INDICATE Rm IS GREATER THAN Rn AS DATA WITHOUT SIGNS,OTHERWISE RESET CONDITIONAL FLAG
cmplo Rm,Rn	<	SET CODITIONAL FLAG WHEN RESULT OF CMP INSTRUCTION INDICATE Rm IS SMALLER THAN Rn AS DATA WITHOUT SIGNS,OTHERWISE RESET CONDITIONAL FLAG
bt label	–	BRANCH WHEN CONDITIONAL FLAG IS SET
movt Rm,Rn	–	TRANSFER Rm TO Rn WHEN CONDITIONAL FLAG IS SET
addt Rm,Rn,Rd	–	ADD Rm AND Rn AND STORE ADDITION RESULT IN Rd WHEN CONDITIONAL FLAG IS SET

Fig. 15

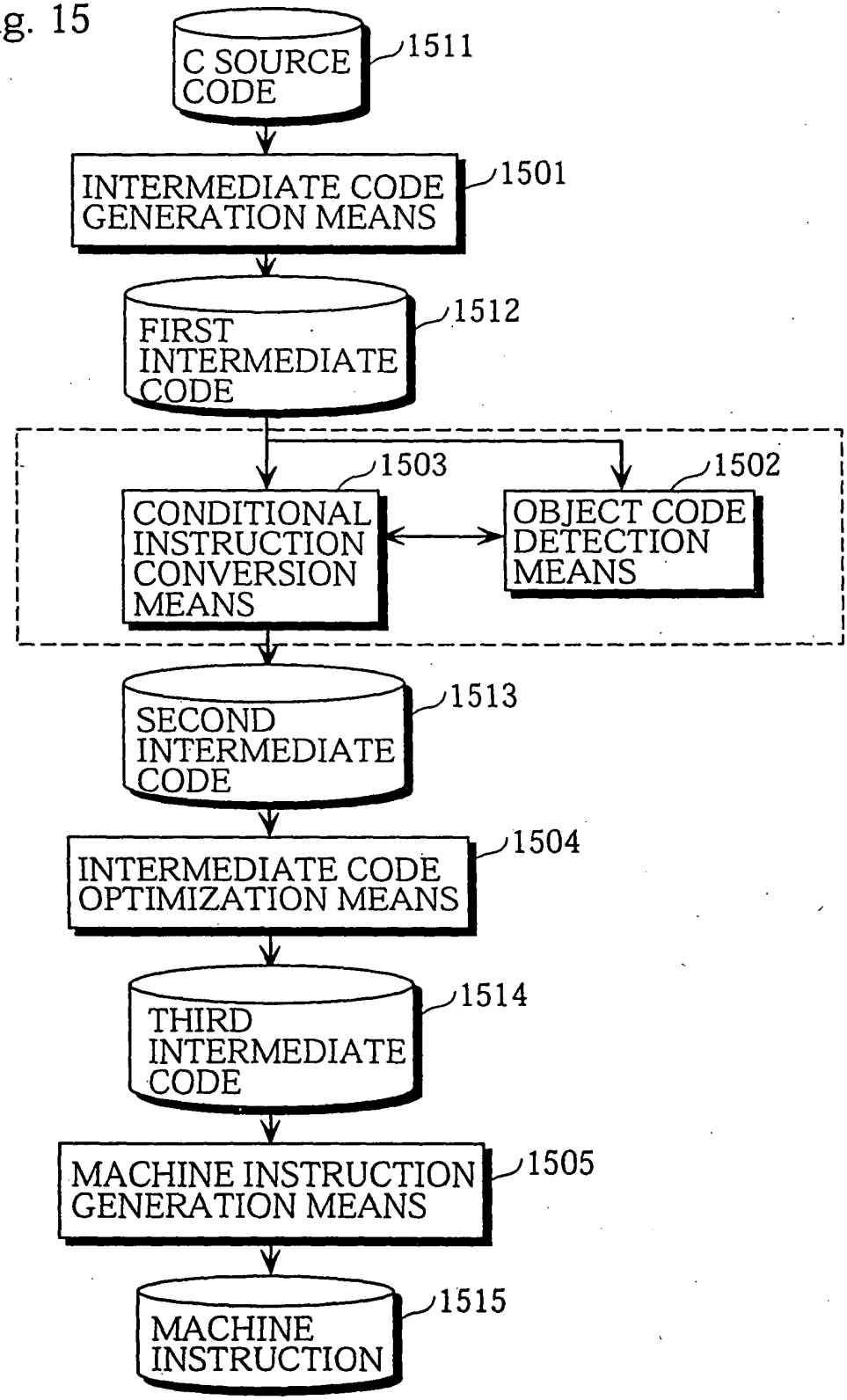
C SOURCE CODE
1511
INTERMEDIATE CODE GENERATION MEANS
1501
FIRST INTERMEDIATE CODE
1512
1503
1502
CONDITIONAL INSTRUCTION CONVERSION MEANS
OBJECT CODE DETECTION MEANS
SECOND INTERMEDIATE CODE
1513
INTERMEDIATE CODE OPTIMIZATION MEANS
1504
THIRD INTERMEDIATE CODE
1514
MACHINE INSTRUCTION GENERATION MEANS
1505
MACHINE INSTRUCTION
1515

Fig. 16

```
if(a == b)
{
        c = 1;
}
else
{
        c = 0;
}
f();
```

Fig. 17

```
        a cmp b
        beq Lt
        c=0
        jmp  L
Lt:     c=1
L:      jsr  f
```

Fig. 18

```
a cmpeq b       ←1801
c=0             ←1802
c=:true 1       ←1803
jsr  f          ←1804
```

Fig. 19

```
cmpeq   r0,r1       ←1901
mov     0,r2        ←1902
movt    1,r2        ←1903
jsr     f           ←1904
```

Fig. 20

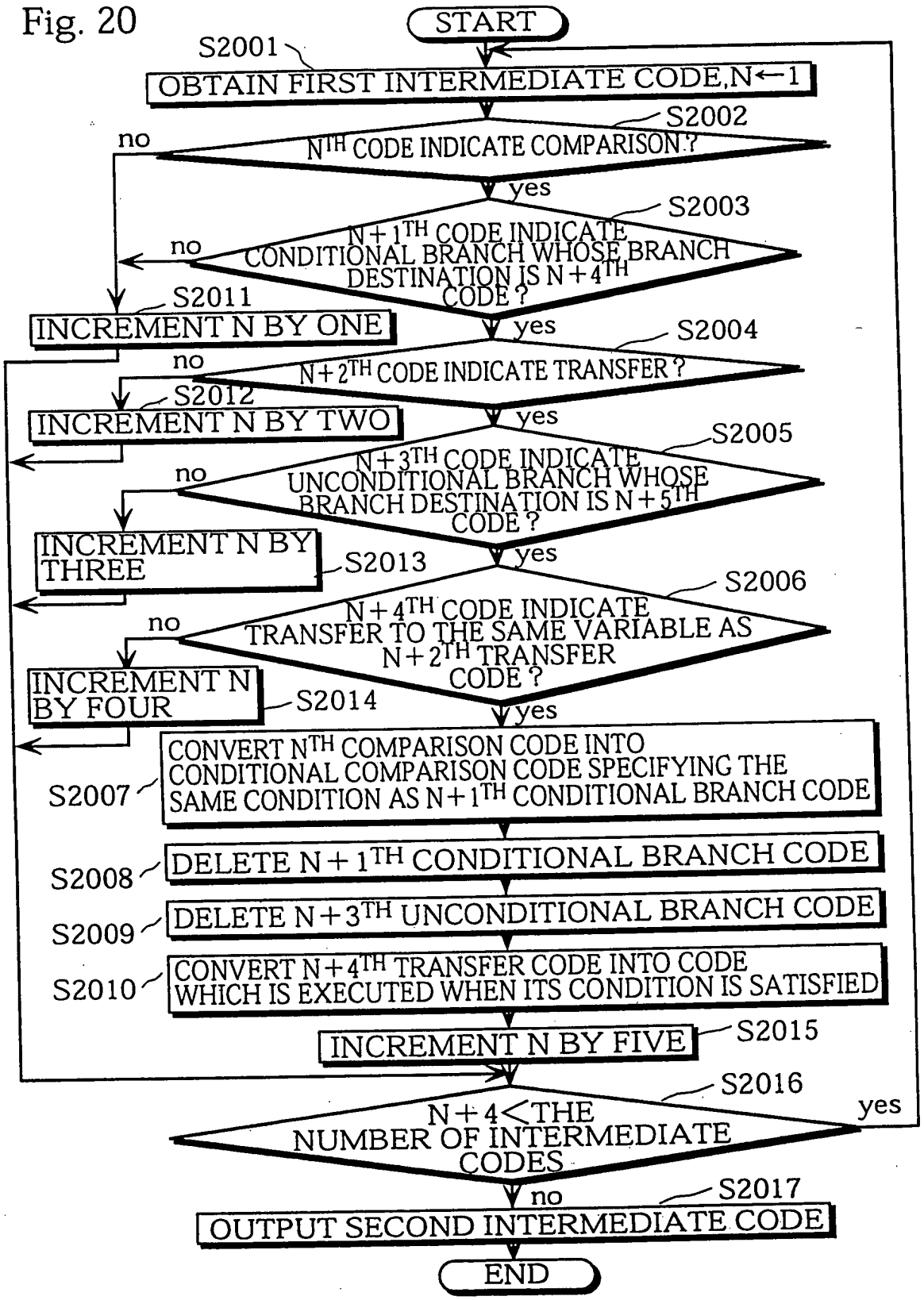
START
S2001
OBTAIN FIRST INTERMEDIATE CODE,N←1
S2002
N^TH CODE INDICATE COMPARISON?
no
yes
S2003
N+1^TH CODE INDICATE CONDITIONAL BRANCH WHOSE BRANCH DESTINATION IS N+4^TH CODE?
no
yes
S2011
INCREMENT N BY ONE
S2004
N+2^TH CODE INDICATE TRANSFER?
no
yes
S2012
INCREMENT N BY TWO
S2005
N+3^TH CODE INDICATE UNCONDITIONAL BRANCH WHOSE BRANCH DESTINATION IS N+5^TH CODE?
no
yes
INCREMENT N BY THREE
S2013
S2006
N+4^TH CODE INDICATE TRANSFER TO THE SAME VARIABLE AS N+2^TH TRANSFER CODE?
no
yes
INCREMENT N BY FOUR
S2014
S2007
CONVERT N^TH COMPARISON CODE INTO CONDITIONAL COMPARISON CODE SPECIFYING THE SAME CONDITION AS N+1^TH CONDITIONAL BRANCH CODE
S2008
DELETE N+1^TH CONDITIONAL BRANCH CODE
S2009
DELETE N+3^TH UNCONDITIONAL BRANCH CODE
S2010
CONVERT N+4^TH TRANSFER CODE INTO CODE WHICH IS EXECUTED WHEN ITS CONDITION IS SATISFIED
INCREMENT N BY FIVE
S2015
S2016
N+4<THE NUMBER OF INTERMEDIATE CODES
yes
no
S2017
OUTPUT SECOND INTERMEDIATE CODE
END

Fig. 21

```
a cmp b
c=0
c=:eq 1
jsr   f
```

Fig. 22

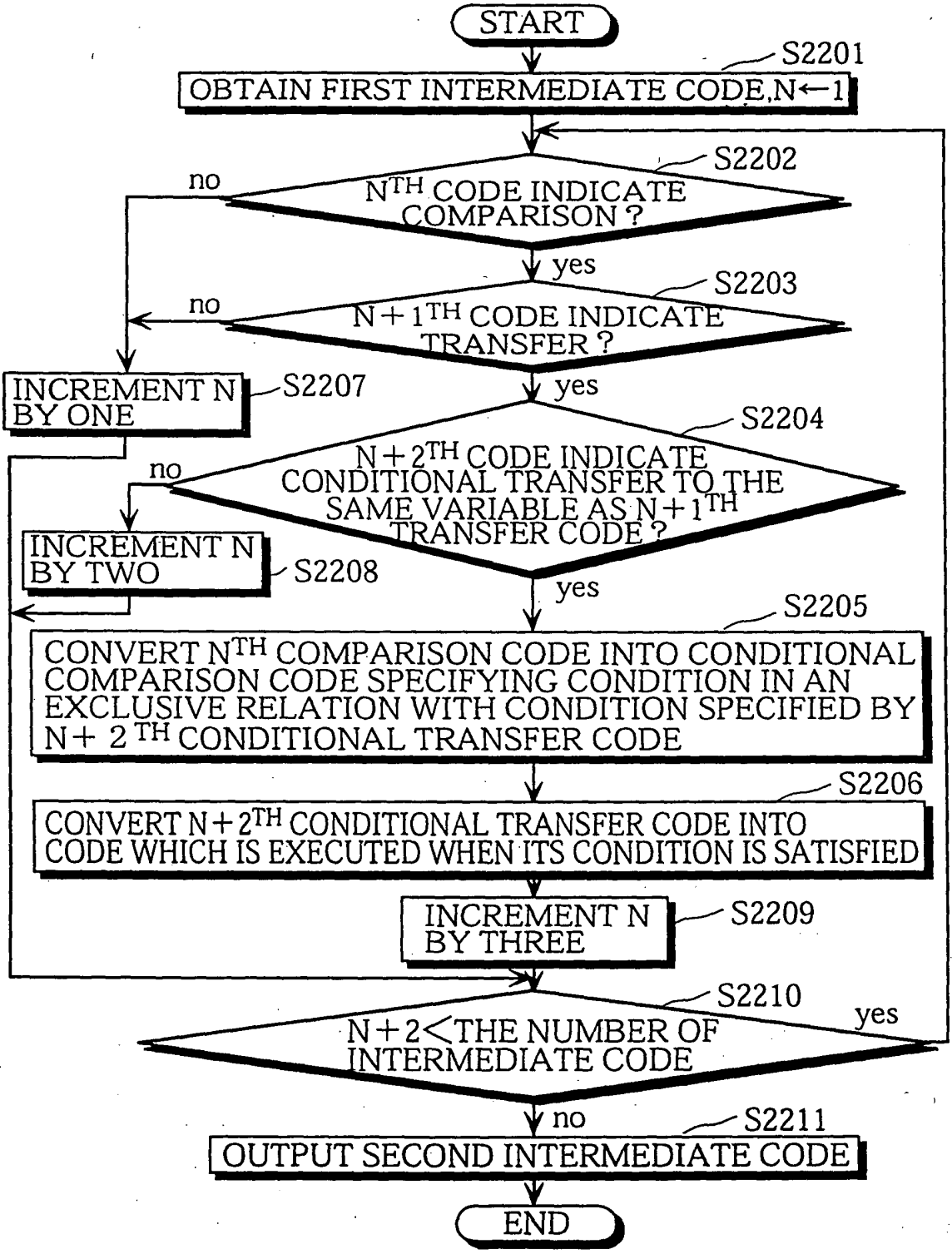
START
S2201
OBTAIN FIRST INTERMEDIATE CODE,N←1
S2202
N^TH CODE INDICATE COMPARISON ?
no
yes
S2203
N+1^TH CODE INDICATE TRANSFER ?
no
yes
INCREMENT N BY ONE
S2207
S2204
N+2^TH CODE INDICATE CONDITIONAL TRANSFER TO THE SAME VARIABLE AS N+1^TH TRANSFER CODE ?
no
yes
INCREMENT N BY TWO
S2208
S2205
CONVERT N^TH COMPARISON CODE INTO CONDITIONAL COMPARISON CODE SPECIFYING CONDITION IN AN EXCLUSIVE RELATION WITH CONDITION SPECIFIED BY N+ 2 ^TH CONDITIONAL TRANSFER CODE
S2206
CONVERT N+2^TH CONDITIONAL TRANSFER CODE INTO CODE WHICH IS EXECUTED WHEN ITS CONDITION IS SATISFIED
INCREMENT N BY THREE
S2209
S2210
N+2<THE NUMBER OF INTERMEDIATE CODE
yes
no
S2211
OUTPUT SECOND INTERMEDIATE CODE
END

Fig. 23

```
a cmp b      ←2301
c=:ne 0      ←2302
c=:eq 1      ←2303
jsr  f       ←2304
```

Fig. 24

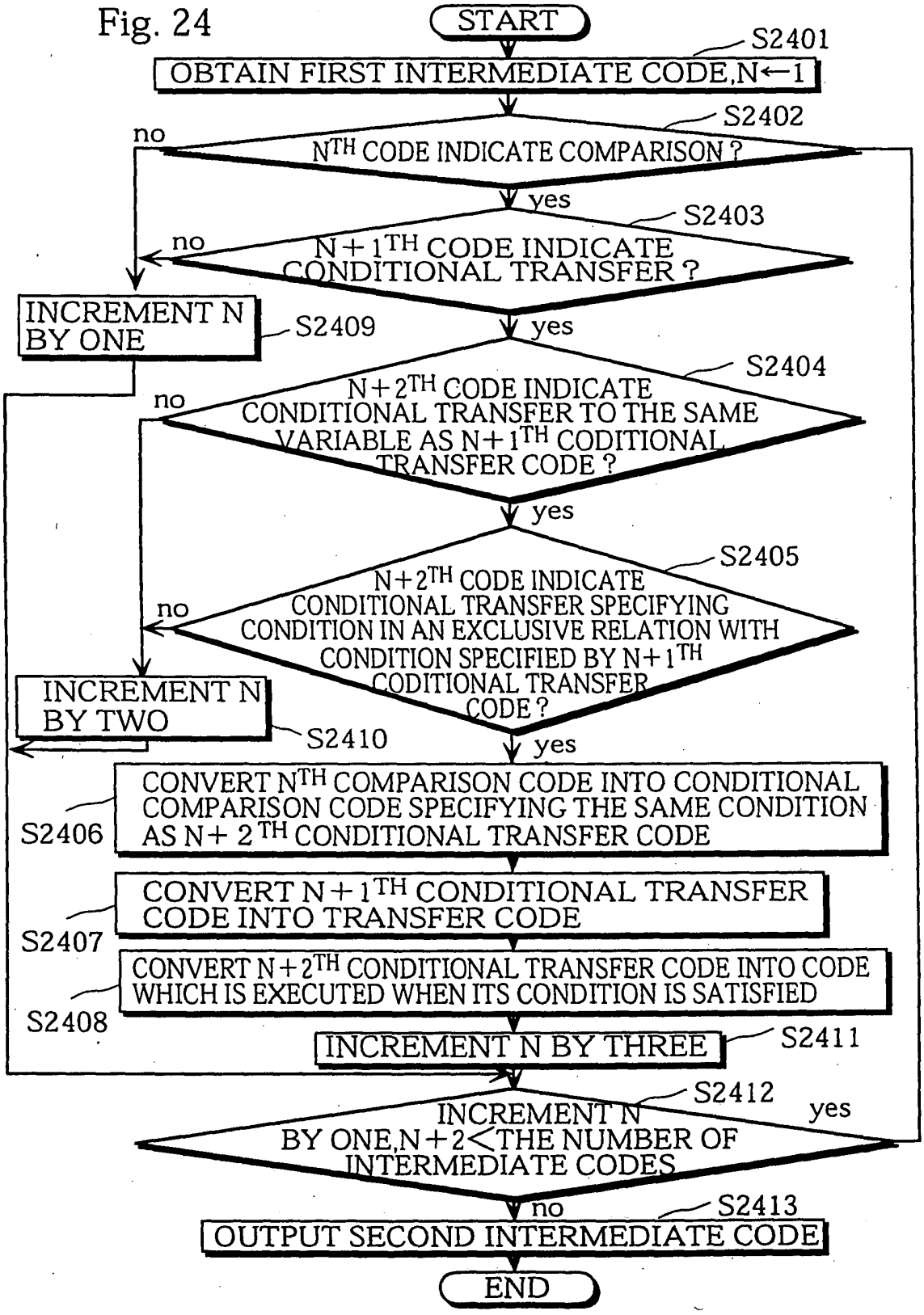
START
S2401
OBTAIN FIRST INTERMEDIATE CODE,N←1
S2402
no
N^TH CODE INDICATE COMPARISON ?
yes
S2403
no
N+1^TH CODE INDICATE CONDITIONAL TRANSFER ?
INCREMENT N BY ONE
S2409
yes
S2404
N+2^TH CODE INDICATE CONDITIONAL TRANSFER TO THE SAME VARIABLE AS N+1^TH CODITIONAL TRANSFER CODE ?
no
yes
S2405
N+2^TH CODE INDICATE CONDITIONAL TRANSFER SPECIFYING CONDITION IN AN EXCLUSIVE RELATION WITH CONDITION SPECIFIED BY N+1^TH CODITIONAL TRANSFER CODE ?
no
INCREMENT N BY TWO
S2410
yes
S2406
CONVERT N^TH COMPARISON CODE INTO CONDITIONAL COMPARISON CODE SPECIFYING THE SAME CONDITION AS N+2^TH CONDITIONAL TRANSFER CODE
S2407
CONVERT N+1^TH CONDITIONAL TRANSFER CODE INTO TRANSFER CODE
S2408
CONVERT N+2^TH CONDITIONAL TRANSFER CODE INTO CODE WHICH IS EXECUTED WHEN ITS CONDITION IS SATISFIED
INCREMENT N BY THREE
S2411
S2412
INCREMENT N BY ONE,N+2<THE NUMBER OF INTERMEDIATE CODES
yes
no
S2413
OUTPUT SECOND INTERMEDIATE CODE
END

Fig. 25

CONDITIONAL OPERATION INSTRUCTION	←2501	CONDITION←2502
addeq	←2503	=
addgt	←2504	>
addge	←2505	≧

Fig. 26

```
if(a  != b)
{
        d = c + 1;
}
else
{
        d = c + 2;
}
f ();
```

Fig. 27

```
        a cmp b
        bne  Lt
        d =  c + 2
        jmp L
Lt:     d =  c + 1
L:      jsr  f
```

Fig. 28

```
a cmp b
d =  c + 1
d =  c +:eq 2
jsr  f
```

Fig. 29

```
cmp     r0,r1
add     1,r2,r3
addeq   2,r2,r3
```

Fig. 30

```
mov     1,r0     ←3001
cmp     r1,r2    ←3002
moveq   0,r0     ←3003
```

Fig. 31

CONDITIONAL TRANSFER INSTRUCTION 3101	CONDITION 3102
moveq	=
movne	≠
movgt	>
movge	≧
movlt	<
movle	≦

Fig. 32

INSTRUCTION IN MNEMONIC CODE [COMPARISON]	SPECIFIED CONDITION	OPERATION
cmp Ra,Rb		COMPARE Ra AND Rb AND SET OPERATION FLAG TO INDICATE COMPARISON RESULT
[CONDITIONAL ADDITION]		
addeq Rd,Rn,Rm	=	ADD Rm AND Rn AND STORE ADDITION RESULT IN Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra AND Rb ARE EQUAL
addne Rd,Rn,Rm	≠	ADD Rm AND Rn AND STORE ADDITION RESULT IN Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra AND Rb ARE NOT EQUAL
addge Rd,Rn,Rm	≧	ADD Rm AND Rn AND STORE ADDITION RESULT IN Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS GREATER THAN OR EQUAL TO Rb AS DATA WITH SIGNS
addle Rd,Rn,Rm	≦	ADD Rm AND Rn AND STORE ADDITION RESULT IN Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS SMALLER THAN OR EQUAL TO Rb AS DATA WITH SIGNS
addgt Rd,Rn,Rm	>	ADD Rm AND Rn AND STORE ADDITION RESULT IN Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS GREATER THAN Rb AS DATA WITH SIGNS
addlt Rd,Rn,Rm	<	ADD Rm AND Rn AND STORE ADDITION RESULT IN Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS SMALLER THAN Rb AS DATA WITH SIGNS
addhs Rd,Rn,Rm	≧	ADD Rm AND Rn AND STORE ADDITION RESULT IN Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS GREATER THAN OR EQUAL TO Rb AS DATA WITHOUT SIGNS
addls Rd,RnRm	≦	ADD Rm AND Rn AND STORE ADDITION RESULT IN Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS SMALLER THAN OR EQUAL TO Rb AS DATA WITHOUT SIGNS
addhi Rd,Rn,Rm	>	ADD Rm AND Rn AND STORE ADDITION RESULT IN Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS GREATER THAN Rb AS DATA WITHOUT SIGNS
addlo Rd,Rn,Rm	<	ADD Rm AND Rn AND STORE ADDITION RESULT IN Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS SMALLER THAN Rb AS DATA WITHOUT SIGNS
[CONDITIONAL TRANSFER]		
moveq Rd,Rm	=	TRANSFER Rm TO Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra AND Rb ARE EQUAL
movne Rd,Rm	≠	TRANSFER Rm TO Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra AND Rb ARE NOT EQUAL
movge Rd,Rm	≧	TRANSFER Rm TO Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS GREATER THAN OR EQUAL TO Rb AS DATA WITH SIGNS
movle Rd,Rm	≦	TRANSFER Rm TO Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS SMALLER THAN OR EQUAL TO Rb AS DATA WITH SIGNS
movgt Rd,Rm	>	TRANSFER Rm TO Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS GREATER THAN Rb AS DATA WITH SIGNS
movlt Rd,Rm	<	TRANSFER Rm TO Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS SMALLER THAN Rb AS DATA WITH SIGNS
movhs Rd,Rm	≧	TRANSFER Rm TO Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS GREATER THAN OR EQUAL TO Rb AS DATA WITHOUT SIGNS
movls Rd,Rm	≦	TRANSFER Rm TO Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS SMALLER THAN OR EQUAL TO Rb AS DATA WITHOUT SIGNS
movhi Rd,Rm	>	TRANSFER Rm TO Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS GREATER THAN Rb AS DATA WITHOUT SIGNS
movlo Rd,Rm	<	TRANSFER Rm TO Rd WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS SMALLER THAN Rb AS DATA WITHOUT SIGNS
[CONDITIONAL BRANCH]		
beq label	=	BRANCH TO label WHEN RESULT OF CMP INSTRUCTION INDICATE Ra AND Rb ARE EQUAL
bne label	≠	BRANCH TO label WHEN RESULT OF CMP INSTRUCTION INDICATE Ra AND Rb ARE NOT EQUAL
bge label	≧	BRANCH TO label WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS GREATER THAN OR EQUAL TO Rb AS DATA WITH SIGNS
ble label	≦	BRANCH TO label WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS SMALLER THAN OR EQUAL TO Rb AS DATA WITH SIGNS
bgt label	>	BRANCH TO label WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS GREATER THAN RN AS DATA WITH SIGNS
blt label	<	BRANCH TO label WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS SMALLER THAN RN AS DATA WITH SIGNS
bhs label	≧	BRANCH TO label WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS GREATER THAN OR EQUAL TO Rb AS DATA WITHOUT SIGNS
bls label	≦	BRANCH TO label WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS SMALLER THAN OR EQUAL TO Rb AS DATA WITHOUT DATA
bhi label	>	BRANCH TO label WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS GREATER THAN Rb AS DATA WITHOUT SIGNS
blo label	<	BRANCH TO label WHEN RESULT OF CMP INSTRUCTION INDICATE Ra IS SMALLER THAN Rb AS DATA WITHOUT SIGNS

Fig. 33

INSTRUCTION IN MNEMONIC CODE	SPECIFIED CONDITION	OPERATION
[COMPARISON]		
cmp/eq Rm,Rn	=	SET CONDITIONAL FLAG WHEN RESULT OF CMP INSTRUCTION INDICATE Rm AND Rn ARE EQUAL, ,OTHERWISE RESET CONDITIONAL FLAG
cmp/ge Rm,Rn	≧	SET CONDITIONAL FLAG WHEN RESULT OF CMP INSTRUCTION INDICATE Rm IS GREATER THAN OR EQUAL TO Rn AS DATA WITH SIGNS,OTHERWISE RESET CONDITIONAL FLAG
cmp/gt Rm,Rn	>	SET CONDITIONAL FLAG WHEN RESULT OF CMP INSTRUCTION INDICATE Rm IS GREATER THAN Rn AS DATA WITH SIGNS,OTHERWISE RESET CONDITIONAL FLAG
cmp/hs Rm,Rn	≧	SET CONDITIONAL FLAG WHEN RESULT OF CMP INSTRUCTION INDICATE Rm IS GREATER THAN OR EQUAL TO Rn AS DATA WITHOUT SIGNS,OTHERWISE RESET CONDITIONAL FLAG
cmp/hi Rm,Rn	>	SET CONDITIONAL FLAG WHEN RESULT OF CMP INSTRUCTION INDICATE Rm IS GREATER THAN Rn AS DATA WITHOUT SIGNS,OTHERWISE RESET CONDITIONAL FLAG
[CONDITIONAL ADDITION]		
addt Rd,Rn,Rm	–	ADD Rm AND Rn AND STORE ADDITION RESULT IN Rd WHEN CONDITIONAL FLAG IS SET
addf Rd,Rn,Rm	–	ADD Rm AND Rn AND STORE ADDITION RESULT IN Rd WHEN CONDITIONAL FLAG IS RESET
[CONDITIONAL TRANSFER]		
movt Rd,Rm	–	TRANSFER Rm TO Rd WHEN CONDITIONAL FLAG IS SET
movf Rd,Rm	–	TRANSFER Rm TO Rd WHEN CONDITIONAL FLAG IS RESET
[CONDITIONAL BRANCH]		
bt label	–	BRANCH WHEN CONDITIONAL FLAG IS SET
bf label	–	BRANCH WHEN CONDITIONAL FLAG IS RESET